REMARKS

Claims 1, 2, 4, 7-21, 23-41, and 44-55 are pending in the application. These same claims also remain rejected. Claims 1, 19, 35 and 40 are the independent claims. The outstanding claims are all variously rejected under 35 U.S.C § 102(e) with reference to U.S. Patent No. 6,401,084 (Ortega et al.) and under 35 U.S.C § 103(a) over root reference Ortega et al., in various combination with U.S. Publication No. 2003/0037077 (Brill et al.), U.S. Publication No. 2002/0059204 (Harris) and U.S. Publication No. 2001/0032204 (Hoashi et al.).

First of all, the Applicants thank the Examiner for the feedback regarding the cited references, however, Applicants maintain that the outstanding rejections are respectfully traversed.

Rejection of Independent Claims 1, 19, 35, and 40 Under 35 U.S.C. § 102(e) and Under § 103(a)

As mentioned, claims 1, 19, 35 and 40 are the independent claims. For example, claim 1 recites:

> In a computing system, a method for providing runtime automatic spelling analysis and correction in connection with a service utilizing a query input mechanism, comprising:
> receiving from a client computing device original query entry data comprising at least one word;
> analyzing the spelling of the at least one word and determining, **for each word,** whether the at least one word has a mistake; and
> forming auto-corrected query entry data wherein said forming includes, for each word having a mistake, replacing the word having the mistake with an alternative word, if the alternative word satisfies at least one threshold confidence calculation.

(emphasis added). Support for amending the claim to recite analyzing the spelling of the at least one word, and then determining **"for each word"** whether the at least one word has a mistake, can be found in the specification:

> Figure 4A illustrates an exemplary embodiment of the spelling check/correct component of the present invention in connection with a search engine service. At 400, a user enters a query on the client computing device. At 405, the client computing device request that the server provide a service, such as search engine results, in connection with the input query. At 410, SSO 350 sends a query to the runtime spell checker 360. *At 415, runtime spell checker 360 breaks the query into a list of words, if there are a plurality of words in the query. At 420, the runtime spell checker 360 examines the first word in the list of words generated at 415. At 425, a loop begins* ***for each word*** *in the list of words. If there is* ***a word*** *to process, at 430,* ***the word*** *is checked against the lexicon sources e.g., the native language and Web lexicons. If* ***the word*** *is in one of the lexicons, then at 435 a pointer to the various words in the list of words is moved to* ***the next word*** *in the list, and the flow returns to 425 to determine if any further processing of words remains.* If, at 430, the word is not in any of the lexicons, then at 440, the possibility of spelling mistake is recognized. Thus, at 440, an algorithm is implemented that determines whether there is a single high confidence correction for the word. If not, the flow proceeds to 435 where the pointer to the list of words is moved to the next word in the list and then to 425 again. If there is a single high confidence correction for the word, then at 445 the word is changed to the corrected word before proceeding to 435 where the pointer to the list of words is moved to the next word in the list and to 425 thereafter.

(Specification, p. 16, ll. 26 to p. 17 ll.14) (emphasis added). *See Also* Fig. 4A of the Application.

Contrast this to Ortega et al. which uses multiple-terms to perform its form of spelling correction. What Ortega et al. actually does is leverage an *already* correct spelling of one term (the matching term—it is already correct, that's why it's called a "matching term") to get at the correct spelling of second term (the non-matching term) by using related terms to the first term (the matching term) and seeing if those related terms are "sufficiently similar" with the second term (the non-matching term) (col. 1, l. 66 to col. 2, l. 47). Thus, Ortega et al. uses the *already* correct spelling of one query term to obtain a "sufficiently similar" correct spelling of another query term, and as such, it does not teach "analyzing the spelling of the at least one word and determining**, for each word,** whether the at least one word has a mistake" (claim 1) (emphasis added). In fact, Ortega et al. cannot do what claim 1 recites "**for each word**" since Ortega et al. needs at least one search term to leverage off of—the matching term that is matched to the related terms.

Claims 19 and 40 recite similar limitation to that of claim 1: "analyzing the spelling of the at least one word of the original query entry data and determining**, for each word,** whether at least one word has a mistake" (claim 19) (emphasis added);

"means for analyzing the spelling of the at least one word and means for determining, **for each word,** whether at least one word has a mistake;" (claim 40) (emphasis added).

Next, claim 35 recites:

> In a computing system, a method for displaying results of a runtime service based upon an auto-corrected query data set, wherein the auto-corrected query data is different than an entered query data set input to a query input mechanism, comprising:
>
> first displaying the auto-corrected query data set in the query input mechanism;
>
> second displaying the search results based upon the auto-corrected query data set; and
>
> near the query input mechanism, third displaying a link which enables the re-performance of the service with the ***entered query data set.***

(emphasis added). The "link ... enables ... the ***entered query data set***" to be re-performed. The entered query data set is "different" from the auto-corrected query data set (see claim 35).

Contrast this to Ortega et al. The Examiner points to col. 5, ll. 35-43 as allegedly rebutting the Applicants' argument that "Ortega does not teach a link for a user to select as a query input mechanism" (Office Action, p. 16). What col. 5, ll. 35-43 discloses is providing a hyperlink to ***candidate replacement*** terms—not to the originally entered query data sets. Thus, Ortega et al. fails to disclose the limitation of "near the query input mechanism, third displaying a link which enables the re-performance of the service with the ***entered query data set***" (claim 35) (emphasis added).

None of the other references cited by the Examiner, namely, Brill et al., Harris, and Hoashi et al. speak to the limitations in independent claims 1, 19, 35, and 40 discussed above. Nor does the Examiner cite these reference to address these limitations.

In the end, independent claims 1, 19, 35, and 40 recite limitations which cannot be found in the cited references, either alone or in combination. Claims 2, 4, 7-18, 20-21, 23-34,

36-39, 41 and 44-55 depend from claims 1, 19, 35 or 40 and are believed to be allowable for the same reasons. Applicants thus submit that claims 1-2, 4, 7-21, 23-41 and 44-55 patentably define over Ortega et al., taken alone or in combination with any other art of record. Withdrawal of the rejections to claims 1-2, 4, 7-21, 23-41 and 44-55 under 35 U.S.C. §§ 102(e) and 103(a) is thus earnestly requested.

CONCLUSION

Applicants believe that the present Amendment is responsive to each of the points raised by the Examiner in the Office Action, and submit that claims 1-2, 4, 7-21, 23-41 and 44-55 of the application are in condition for allowance. Favorable consideration and passage to issue of the application at the Examiner's earliest convenience is earnestly solicited.

Date: May 24, 2005

Grzegorz S. Plichta
Registration No. 55,541

Woodcock Washburn LLP
One Liberty Place - 46th Floor
Philadelphia PA 19103
Telephone: (215) 568-3100
Facsimile: (215) 568-3439